

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-108260

(43) Date of publication of application : 11.04.2003

(51)Int.Cl. G06F 1/04
G06F 1/32

(21)Application number : 2001-302429 (71)Applicant : HITACHI LTD

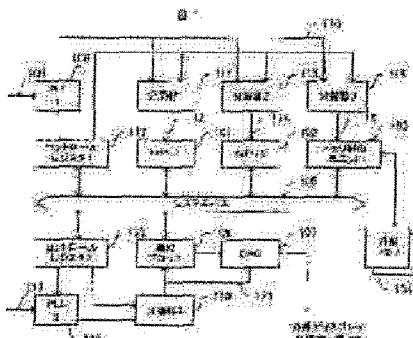
(22)Date of filing : 28.09.2001 (72)Inventor : KUROKAWA YOSHITAKE
NISHIOKA KIYOKAZU
NAKADA KEIMEI
HOSOKI KOJI

(54) INFORMATION PROCESSER AND CONTROLLING METHOD THEREFOR

(57) Abstract:

PROBLEM TO BE SOLVED: To allow changing an operating frequency and a voltage enabling to operate at the minimum consumption power possible to achieve a processing capacity enough to perform an application.

SOLUTION: An apparatus for information processing comprises MPUs 101 and 102, a memory control unit 103, dividers 111, 113, 115 and 120 whose frequency-division ratios are individually controlled to a functional block 106 and vary those operating frequencies. An application on the apparatus prepares information on operating frequencies optimum to each MPU 101 and 102 and the functional block 106. A control means for controlling the information for a plurality of applications is prepared to acquire the present operating status from the control means when initiating the applications and allocates the processing capacity needed by the applications to be initiated to each MPU 101 and 102 and the functional block 106.



to vary those operating frequencies, and the same operation is performed when exiting the applications.

LEGAL STATUS

[Date of request for examination] 11.08.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

〔19〕日本国特許序 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2003-108260
(P2003-108260A)

(43)公開日 平成15年4月11日(2003.4.11)

(51) Int.Cl. 7 藏別記号
G 06 F 1/04 301
1/32

F I 7-3-1-(参考)
G 0 6 F 1/04 3 0 1 C 5 B 0 1 1
1/00 3 3 2 Z 5 B 0 7 9

審査請求 未請求 請求項の数? 01 (全 15 頁)

(21)出願番号 特願2001-302429(P2001-302429)

(22)出願日 平成13年9月28日(2001.9.28)

(71)出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目 6 番地

(72)発明者 黒川 能毅
神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72)発明者 西岡 清和
神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(74)代理人 100093492
弁理士 鈴木 市郎 (外1名)

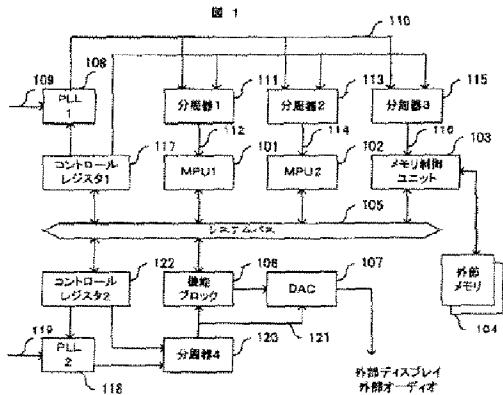
最終頁に統ぐ

(54) 【発明の名称】 情報処理装置及び情報処理装置の制御方法

(57) 【類約】

【課題】 アプリケーションに対して充分な処理能力が実現できる最小の消費電力で動作できる様に動作周波数、動作電圧を変更可能とする。

【解決手段】 MPU101、102、メモリ制御ユニット103、機能ブロック106に対して、独立に分周比が制御される分周器111、113、115、120を持ち、それらの動作周波数を変更する。また、情報処理装置上のアプリケーションが各MPU、機能ブロックに最適な動作周波数情報を用意する。そして、複数のアプリケーションに対しこの情報を管理する手段を用意し、アプリケーションを開始したとき、現在の動作状況を管理手段から取得し、開始するアプリケーションに必要な処理能力を各MPU、機能ブロックに割り振り、それらの動作周波数を変更する。アプリケーション終了時にも同様の動作を行う。



のそれぞれが持つプロセッサ、機能ブロックのそれぞれに最適な動作周波数情報を基づいて、前記プロセッサ、機能ブロックの動作周波数を決定し、前記プロセッサ、機能ブロックのそれぞれの動作周波数を独立に変更することを特徴とする情報処理装置の制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理装置及び情報処理装置の制御方法に係り、特に、動作周波数を変更可能とした情報処理装置及び情報処理装置の制御方法に関する。

【0002】

【従来の技術】近年、信号処理やメディア処理等の分野において、プロセッサや専用ハードウェアを使用した処理が行われるようになってきた。これらの処理は、その演算量が非常に大きく、また、一定時間内の処理終了が必須となるリアルタイム性が要求される。このため、これらの処理を実現する情報処理装置は、非常に高い処理性能が要求され、これを実現するために高い動作周波数が要求されている。

【0003】一般に、プロセッサ等のCMOSトランジスタで構成されている半導体回路は、その動作周波数に比例して消費電力が増加する。これらの消費電力は、情報処理装置を構成する際の障害となるため、消費電力を低減するための種々の方法が提案されている。

【0004】例えば、マルチプロセッサンシステム等における消費電力の低減を図ることを可能にする従来技術として、例えば、特開平8-202468号公報等に記載された技術が知られている。この従来技術は、マルチプロセッサ構成のシステムにおいて、各プロセッサについて処理状態を検出し、使用していないかまたは負荷の軽いプロセッサを低消費電力モードに移行させ、システム全体の消費電力を押さえるというものである。

【0005】また、端末の従来技術として、例えば、特開2000-66776号公報等に記載された技術が知られている。この従来技術は、システムのサブ回路の電力消費を制御する方法に関するもので、プロセッサに割り当てられた負荷に見合った動作周波数を割り出し、この周波数でプロセッサが動作することができる電源電圧を検出し、この電源電圧をプロセッサに印加することによって、その負荷を処理できる最小電力でプロセッサを動作させることを可能とするものである。

【0006】前述したように、従来からマルチプロセッサシステムのような大規模な情報処理装置を構成するために、全体の消費電力を減らすという試みがなされてきた。

【0007】一方、半導体回路の集積度が年々上がっており、この結果、以前では複数の部品で構成されていた情報処理装置を1つの半導体回路に集積して構成することが可能となってきた。これによって、より小さな

【特許請求の範囲】

【請求項1】1または複数のプロセッサ、専用論理を持つ1または複数の機能ブロックのいずれか1方あるいは両方を備える情報処理装置において、前記プロセッサ、機能ブロックのそれぞれの動作周波数を独立に変更する手段と、情報処理装置上で動作する1または複数のアプリケーションのそれぞれが持つプロセッサ、機能ブロックのそれそれぞれに最適な動作周波数情報を基づいて、前記プロセッサ、機能ブロックの動作周波数を決定し、前記動作周波数を変更する手段を制御する制御手段とを備えることを特徴とする情報処理装置。

【請求項2】前記制御手段は、アプリケーションが開始されたとき、現在の情報処理装置の動作状況を管理している管理手段から動作状況を取得し、開始するアプリケーションに必要な処理能力をプロセッサ、機能ブロックに割り振り、前記プロセッサ、機能ブロックの動作周波数を変更することを特徴とする請求項1記載の情報処理装置。

【請求項3】前記制御手段は、アプリケーションが終了されたとき、現在の情報処理装置の動作状況を管理している管理手段から動作状況を取得し、終了するアプリケーションに割り当てられていた処理能力をプロセッサ、機能ブロックから削除して、前記プロセッサ、機能ブロックの動作周波数を変更することを特徴とする請求項1記載の情報処理装置。

【請求項4】バッテリ等の主電源と、その電圧を測定する測定手段とをさらに備え、前記制御手段は、前記プロセッサ、機能ブロックの動作周波数を、前記測定手段で測定された電源電圧で動作可能な周波数に設定することを特徴とする請求項1、2または3記載の情報処理装置。

【請求項5】外部からのフォーマットされたデータ入力を受け付ける手段さらに備え、前記制御手段は、前記プロセッサ、機能ブロックの動作周波数を、前記データ入力のデータフォーマットに応じて、このデータフォーマットに適した動作周波数に設定することを特徴とする請求項1ないし4のうちいずれか1記載の情報処理装置。

【請求項6】電源電圧を変更する手段をさらに備え、前記制御手段は、現在動作している1または複数のアプリケーションが必要としている動作周波数の総和で、前記前記プロセッサ、機能ブロックが動作することができるような電圧を、前記前記プロセッサ、機能ブロックに印加するように前記電源電圧を変更する手段を制御することを特徴とする請求項1ないし5のうちいずれか1記載の情報処理装置。

【請求項7】1または複数のプロセッサ、専用論理を持つ1または複数の機能ブロックのいずれか1方あるいは両方を備える情報処理装置の制御方法において、情報処理装置上で動作する1または複数のアプリケーション

情報処理装置の開発が可能となり、様々な小型機器への応用が行われている。特に、バッテリ等の電力供給源を使用することによって、携帯可能な情報処理装置の作成が可能となっている。従来、このような用途で半導体回路が使用される場合、電力供給源の容量に見合った最大消費電力を持つものを開発または選択してきたが、この方法は、前述したメディア処理等の高負荷の処理を行う情報処理装置を実現することが難しい。

【0008】

【発明が解決しようとする課題】前述したように、情報処理装置は、処理能力の増加が要求される一方で、バッテリ動作が可能である等の省電力化が要求されている。このため、情報処理装置には、これらの2つの要求を満たすために、ある情報処理装置が処理すべき対象であるアプリケーションに対して十分な処理能力を有し、かつ、その処理能力が実現できる最小の消費電力で動作することが要求されている。

【0009】しかし、前述した従来技術による情報処理装置は、前述したような2つの要求を同時に、かつ、充分に満足させることができるものではないという問題点を有している。

【0010】本発明の目的は、アプリケーションに対して充分な処理能力が実現できる最小の消費電力で動作できるように、動作周波数、動作電圧を変更することのできる高い処理能力を持ち、低消費電力化を図った情報処理装置及び情報処理装置の制御方法を提供することにある。

【0011】

【課題を解決するための手段】本発明によれば前記目的は、1または複数のプロセッサ、専用論理を持つ1または複数の機能ブロックのいずれか1方あるいは両方を備える情報処理装置において、前記プロセッサ、機能ブロックのそれぞれの動作周波数を独立に変更する手段と、情報処理装置上で動作する1または複数のアプリケーションのそれぞれが持つプロセッサ、機能ブロックのそれぞれに最適な動作周波数情報に基づいて、前記プロセッサ、機能ブロックの動作周波数を決定し、前記動作周波数を変更する手段を制御する制御手段とを備えることにより達成される。

【0012】また、前記目的は、1または複数のプロセッサ、専用論理を持つ1または複数の機能ブロックのいずれか1方あるいは両方を備える情報処理装置の制御方法において、情報処理装置上で動作する1または複数のアプリケーションのそれぞれが持つプロセッサ、機能ブロックのそれぞれに最適な動作周波数情報に基づいて、前記プロセッサ、機能ブロックの動作周波数を決定し、前記プロセッサ、機能ブロックのそれぞれの動作周波数を独立に変更することにより達成される。

【0013】

【発明の実施の形態】以下、本発明による情報処理装置

の実施形態を図面により詳細に説明する。

【0014】図1は本発明の第1の実施形態による情報処理装置のハードウェア構成を示すブロック図、図2はアプリケーションプログラムの追加して起動したときに使用するプログラムの処理動作を説明するフローチャート、図3はアプリケーションプログラム終了時に使用するプログラムの処理動作を説明するフローチャート、図4は図2の処理で使用するクロックテーブルについて説明する図、図5は図2の処理でアプリケーションが追加10起動された後のクロックテーブルの状態を説明する図、図6は図3処理でアプリケーションが終了させられた後のクロックテーブルの状態を説明する図であり、以下、これらの図を参照して本発明の第1の実施形態について説明する。

【0015】図1において、101、102は第1、第2のマイクロプロセッサユニット（以下、MPUという）、103は外部メモリ制御ユニット、104は外部メモリ、105はシステムバス、106は機能ブロック、107はデジタルアナログコンバータ（以下、DACという）、108、118は第1、第2のフェイズロックループ（以下、PLLという）、109、119は第1、第2のPLL用リファレンスクロック、110は第1のPLLの発振クロック、111、113、115、120は第1～第4の分周器、112、114は第1、第2のMPU用ローカルクロック、116はメモリ制御ユニット用ローカルクロック、117、122は第1、第2のコントロールレジスタ、121は機能ブロック及びDAC用ローカルクロックである。

【0016】図1に示す本発明の第1の実施形態において、第1のPLL108は、第1のコントロールレジスタ117の値を元にリファレンスクロック109を倍化して発振クロック110を作成する。この発振クロック110は、第1～第3の分周器111、113、115を通して、第1のMPU101、第2のMPU102及びメモリ制御ユニット103を駆動するためのローカルクロック112、114、116の生成のために使用される。分周器111、113、115のそれぞれは、第1のコントロールレジスタ117から値に基づいて独立に制御され、発振クロック110を分周して1/nの周波数に落とす。分周器111、113、115を、第1、第2のMPU101、102及びメモリ制御ユニット103の各ユニット毎に備えることより、各ユニットは、その動作周波数を発振クロック110の必要な1/n倍（nの値は分周器毎に異なってよい）に設定された周波数として受けが可能となり、必要とする周波数で動作することができる。

【0017】第1、第2のMPU101、102は、外部メモリ104からプログラム及びデータを読み込み、演算を行い、また、システムバス105を通して機能ブロック106、第1、第2のコントロールレジスタ11

7、122を制御することが可能である。メモリ制御ユニット103は、第1、第2のMPU101、102、機能ブロック106からの要求をシステムバス105を通して受けると、外部メモリ104を制御してデータの読み書きを行い、必要に応じて結果を要求元へ返信する。システムバス105は、これに接続されている全てのユニットに対し、要求元から要求先へ要求を伝え、結果を要求先から要求元へ返す。

【0018】説明している本発明の実施形態は、便宜上、システムバス105及びこれに接続されている各ユニット(101、102、103、106、117、122)の中でシステムバス105を制御している部分の全てが、メモリ制御ユニット103用のローカルクロック116で動作しているものとする。しかし、本発明は、システムバス105を、他のユニットのローカルクロックで動作させるようにしても、あるいは、システムバス専用のローカルクロックを新たに作成して動作させるようにしてもよい。

【0019】第2のPLL118は、第1のPLL108と同様の動作を行う。すなわち、第2のPLL118は、第2のコントロールレジスタ122の値を使用し、リファレンスクロック119を遅倍化して第4の分周器120に出力する。第4の分周器120は、第2のコントロールレジスタ122の値を使用して第4の分周器120からの遅倍化されクロックを分周し、ローカルクロック121を作成し、機能ブロック106及びDAC107に供給する。機能ブロック106は、第1、第2のMPU101、102では処理しきれない計算を行う処理に特化した専用論理により構成されている。そして、機能ブロック106は、第1のMPU101または第2のMPU102によって制御され、デジタルビデオ画像やデジタル音声のフィルタ処理、改ざん防止のためにソフトウェアでの実装が不可能な暗号処理等を行う。

【0020】前述した本発明の第1の実施形態は、第1、第2のMPU101、102に後述するプログラムを動作させ、コントロールレジスタ117、122を制御し、第1、第2のMPU101、102、メモリ制御ユニット103、機能ブロック106の動作周波数を変更することによって、システム全体の消費電力を押さえよう動作する。

【0021】なお、前述した図1に示す情報処理装置の例は、2台のMPUと1台の機能ブロックとを備えるとしているが、MPUと機能ブロックとは、それぞれ1または複数が設けられてよく、また、MPU、機能ブロックの一方だけが設けられてもよい。

【0022】次に、図2、図3を参照して、第1、第2のMPU101、102で使用するプログラムの動作を説明するが、これらのプログラムは、図1に示す外部メモリ104に格納され、必要に応じてMPUに読み込まれて処理される。また、これらのプログラムは、後述す

る図4～図6に示すクロックテーブルを使用するが、このテーブルも外部メモリ104に格納されていて、必要に応じてMPUに読み込まれて処理される。

【0023】図4に示すクロックテーブルは、アプリケーション番号の項目401、第1のMPUのクロック項目402、第2のMPUのクロック項目403、メモリ制御ブロックのクロック項目404、機能ブロック、DAC、ディスプレイのクロック項目405をパラメータとして持つ、これらが、システムのパラメータ欄406、第1のアプリケーションのパラメータ欄407、第2のアプリケーションのパラメータ欄408、総和値のパラメータ欄409、最大許容量のパラメータ欄410に格納されて構成される。また、このテーブルには、各パラメータの判定欄411が設けられる。

【0024】パラメータ欄406～408のクロック項目402～405の数値は、図1に示すリファレンスクロック109または119と、ローカルクロック112、114、116、121との比を示す値、すなわち、第1のPLLによる遅倍値と第1～第3の分周器の分周値との積に相当する値、及び、第2のPLLによる遅倍値と第4の分周器の分周値との積に相当する値であり、リファレンスクロック109または119の周波数にこれらの値を乗じたものが、システム及び第1、第2のアプリケーションが必要とする第1、第2のMPU、メモリ制御ユニット、DACでのクロック周波数を示している。ここで、クロック周波数を示す値が0の場合、対応するアプリケーションが、そのクロック項目のMPU等を使用しないことを意味する。

【0025】また、総和値のパラメータ欄409の各クロック項目の値は、この値にリファレンスクロック109または119の周波数を乗じた値が、システム及び第1、第2のアプリケーションが同時に動作する場合の第1、第2のMPU、メモリ制御ユニット、DACでの必要なクロック周波数となることを示している。さらに、最大許容量のパラメータ欄410の各値は、この値にリファレンスクロック109または119の周波数を乗じた値が、第1、第2のMPU、メモリ制御ユニット、DACが動作可能な最大のクロック周波数を示すことになり、判定欄411には、テーブルの状態で第1、第2のMPU、メモリ制御ユニット、DACが動作可能か否か示される。

【0026】そして、図2により後述する処理が実行され新たに第3のアプリケーションが追加して起動されると、図5に示すように、図4のテーブルに第3のアプリケーションに対するパラメータ欄501が設けられると共に、総和値のパラメータ欄409がアプリケーションの追加により変更され、パラメータ欄502に変更される。また、図3により後述する処理が実行され第2のアプリケーション終了されると、図6に示すように、図5に示すテーブルから第2のアプリケーションに対するパ

ラメータ欄408が削除されると共に、総和値のパラメータ欄がパラメータ欄601のように変更される。

【0027】なお、第1、第2のアプリケーションが必要とする第1、第2のMPU、メモリ制御ユニット、DACでのクロック周波数を決める値は、各アプリケーションが、その内部にそれぞれ有している。

【0028】まず、図2に示すフローを参照して、本発明の実施形態において、アプリケーションプログラムを新たに追加して起動させる場合に使用するプログラムの処理動作を説明する。

【0029】(1) この処理プログラムは、ユーザ等によりプログラムの起動要求が発生した場合にスタートし、まず、新たに起動するアプリケーション用にアプリケーション番号を取得する(ステップ201)。

【0030】(2) 次に、202で外部メモリ104に格納されている図4に示すクロックテーブルに新しい欄を作成し、新アプリケーション番号及びアプリケーションが持つ固有のパラメータ402、403、404、405を格納してクロックテーブルを更新する。図5に示すパラメータ欄501が新たに作成された新アプリケーション用の欄である(ステップ202)。

【0031】(3) ステップ202で更新されたクロックテーブルについて、パラメータ402～405の各パラメータの総和を求め、総和値のパラメータ欄409を更新してパラメータ欄502とする(ステップ203)。

【0032】(4) ステップ203の処理で更新したパラメータ欄502の各パラメータの全てが最大許容量欄410内の値を超えていないか否かを判定し、1つでも超えている場合、ステップ202で作成したクロックテーブルの新しい欄501を削除し、パラメータ値の総和の欄409を元の値に戻してここで処理を終了する。この場合、起動要求があったアプリケーションは、現システム状態では動作が不可能と判定し起動しない(ステップ204、205)。

【0033】(5) ステップ204の判定で、更新したパラメータ欄502の各パラメータの全てが最大許容量欄410内の値を超えていなかった場合、新たな各パラメータ総和値をテーブルに格納して、総和値パラメータ502に更新する(ステップ206)。

【0034】(6) 次に、ステップ206で更新された各パラメータの総和値に基づいて、各PLL、分周器に対するコントロールレジスタの値を変更し、各ローカルクロック112、114、116、121の周波数を設定する。この後、新しいアプリケーション本体を起動する(ステップ207、208)。

【0035】前述した処理の例は、2つのアプリケーションが動作している状態で、クロックテーブルが図4に示すような状態の場合に、もう1つのアプリケーションを追加して起動するとして説明したが、前述の説明から

判るように、システムの立ち上げ時におけるクロックテーブルは、システムのパラメータ欄406、総和値のパラメータ欄409、最大許容量のパラメータ欄410、各パラメータの判定欄411だけが設けられる構成であり、アプリケーションが立ち上げられる度に、それらのアプリケーションに対応するパラメータ欄が追加されていく。

【0036】次に、図3に示すフローを参照して、本発明の実施形態において、アプリケーションプログラムを10終了させる場合に使用するプログラムの処理動作を説明する。ここでの処理は、図2により説明した処理の後に、第2のアプリケーションを終了させるものとした場合の例である。

【0037】(1) この処理プログラムは、ユーザ等によって終了するアプリケーションが指定されると起動され、まず、終了するアプリケーションを使用し図5のクロックテーブルから該当するパラメータ欄を探し出し、次に、そのパラメータ欄をクロックテーブルから削除する(ステップ301、302)。

20 【0038】(2) その後、パラメータ402～405の各パラメータの総和値を再計算して求め、総和値のパラメータ欄502を更新してパラメータ欄601とする(ステップ303)。

【0039】(3) 次に、指定されたアプリケーションプログラムを終了させ、各パラメータの総和値欄601から各PLL、分周器に対するコントロールレジスタの値を更新し、各ローカルクロック112、114、116、121の周波数を所定の周波数に設定して処理を終了する(ステップ304、305)。

30 【0040】前述した本発明の第1の実施形態によれば、各MPU、ハードウェア機能ブロックを適正な動作周波数で動作することができるよう設定することができ、かつ、システム全体の消費電力を抑えることが可能となる。

【0041】図7は本発明の第2の実施形態による情報処理装置のハードウェア構成を示すブロック図、図8はストリームデータの処理を開始するときに使用するプログラムの処理動作を説明するフローチャート、図9は図8の処理でストリームデータの処理が開始された後のクロックテーブルの状態を説明する図であり、以下、これらの図を参照して本発明の第2の実施形態について説明する。図7において、701はストリームデータ入力ブロックであり、他の符号は図1の場合と同一である。

【0042】図7に示す本発明の第2の実施形態は、図1に示す第1の実施形態による情報処理装置にストリームデータ入力ブロック701をシステムバス105に接続して加えて構成し、ストリームデータの処理を可能としたもので、その他の構成は、図1の場合と同一である。第2の実施形態で加えたストリームデータ入力ブロック701は、外部からのデジタル映像、音声等のデー

タ入力を随時受け付け、これをシステムバス105を経由して外部メモリ104に送付する処理を行うものである。

【0043】外部からのデジタル映像、音声等のデータとしては、例えば、システム外部のデジタルTVチューナー、ラジオチューナ等が考えられる。これらのデータを以下ではストリームデータと呼ぶが、このようなストリームデータを処理する場合、発信側でストリームデータを作成したときに使用した形式に沿って処理を行う必要がある。これらの形式は、主にデータ量、発信側の都合等により選択されている。受信側を複数の形式に対応できるように構成した場合、このストリームかどの形式に対応するかを判定することができるよう構成する必要がある。一般的に、この複数の形式は、形式毎に必要とする処理能力が異なる。

【0044】本発明の第2の実施形態は、図8により説明する処理を実行するプログラムを使用することによって、ストリームの形式を変更する度に適切な動作周波数を設定することを可能としたものであり、次に、図8に示すフローを参照して、外部ストリームの処理が開始されたときの処理を説明する。この処理においても、図4により説明したクロックテーブルが使用されるものとし、また、このクロックテーブルの第1のアプリケーションに対するパラメータ欄が、この処理でのストリームデータ処理プログラムに相当するものとする。

【0045】(1) この処理プログラムは、ストリームデータ処理プログラムと同時に起動され、まず、入力ストリーム形式が変更されたか否かの判定を行い、変更されていない場合、一定期間を置いて繰り返し判定を行う(ステップ801)。

【0046】(2) ステップ801の判定で、入力ストリーム形式が変更されていた場合、該当アプリケーションの欄に新しい入力ストリーム形式のパラメータを格納する。この処理によって、図4の407に示した第1のアプリケーションのパラメータ欄は、図9に901として示すアプリケーション1aのパラメータ欄のように変更してテーブルを更新する(ステップ802)。

【0047】(3) 次に、ステップ802で更新されたクロックテーブルについて、パラメータ402~405の各パラメータの総和を求め、総和値のパラメータ欄409を更新してパラメータ欄902とする(ステップ803)。

【0048】(4) ステップ803の処理で更新したパラメータ欄902の各パラメータの全てが最大許容欄410内の値を超えていないか否かを判定し、1つでも超えている場合、ステップ802で更新したクロックテーブルの更新した欄901を削除し、パラメータ値の総和の欄902の各パラメータの総和値をもう一度求め直して、クロックテーブル総和値の欄902を更新し、ここでの処理を終了する(ステップ804、805)。

【0049】(5) ステップ804の判定で、全てのパラメータが最大許容量を超えていなかった場合、ステップ803で更新された総和値に基づいて、各PLL、分周器に対するコントロールレジスタの値を変更し、各ローカルクロック112、114、116、121の周波数を所定の値に設定する。この後、新しいストリーム形式での処理動作を開始し、ステップ801からの処理に戻る(ステップ806、807)。

【0050】前述した本発明の第2の実施形態によれば、第1の実施形態の効果に加え、さらに外部からの入力データのフォーマット変化に対応して動作周波数を変更することが可能となり、システム全体の消費電力を押さえることができる。

【0051】図10は本発明の第3の実施形態による情報処理装置のハードウェア構成を示すブロック図、図11は電源電圧が変動したときに動作周波数を変更するために使用するプログラムの処理動作を説明するフローチャート、図12は図11の処理で動作周波数が変更された後のクロックテーブルの状態を説明する図であり、以下、これらの図を参照して本発明の第3の実施形態について説明する。図10において、1001はバッテリ、1002は電圧計、1003は電源モニタレジスタ、1004は電圧保存レジスタであり、他の符号は図1の場合と同一である。

【0052】図10に示す本発明の第3の実施形態は、電源電圧が低下したときに処理装置全体の動作周波数を低下させて消費電力の低減を図ることを可能にしたもので、バッテリ1001の電圧を測定する電圧計1002と、電圧計1002でのバッテリの測定電圧をモニタしてその信号をシステムバス105を介してシステムに与える電源モニタレジスタと、前回のクロックテーブルの更新時におけるバッテリの電圧値を保存している電圧保存レジスタ1004とが、図1により説明した本発明の第1の実施形態に加えられて構成されており、その他の構成は、図1の場合と同一である。

【0053】通常、バッテリ1001は、電力を消費していくと、電圧が低下していく。電圧計1002は、この電圧の低下を検知し、電源モニタレジスタ1003に随時反映する。このレジスタ1003は、システムバス105を経由し、第1、第2のMPU101、102から参照可能である。

【0054】次に、図11に示すフローを参照して、電源電圧が変動したときに動作周波数を変更する処理について説明する。なお、この処理においても、図4により説明したクロックテーブルが参照される。また、電圧保存レジスタ1004には、前回のクロックテーブルの更新時、すなわち、クロックテーブルが図4に示す状態とされたときのバッテリの電圧値が保存されている。

【0055】(1) この処理プログラムは、システムの起動時に起動され、起動されると、まず、電圧モニタレ

ジスタ1003の値を読み出し、電圧モニタレジスタの値が更新されたか否かを判定する。電圧モニタレジスタ1003には、バッテリ1001の電圧変化が當時反映されていて、変化がない場合、一定間隔で更新されたか否かの判定を続ける（ステップ1101、1102）。

【0056】（2）ステップ1102での判定で、電圧モニタレジスタの値が更新されていた場合、前回のクロックテーブルの更新時のバッテリの電圧値を保存している電圧保存レジスタ1004の値と、更新されている電圧モニタレジスタの値とを比較し、電圧モニタレジスタの値が電圧保存レジスタの値より一定以上大きいか否かを判定する（ステップ1103）。

【0057】（3）ステップ1103での判定で、電圧モニタレジスタの値が電圧保存レジスタの値より一定以上大きくなかった場合、電圧モニタレジスタの値が電圧保存レジスタの値より一定以上小さいか否かを判定し、一定以上小さくなかった場合、すなわち、差が一定値未満の場合、電圧の変化はクロックテーブルの設定値の変更を行うほど大きくなかったと判定し、何も変更せずに、ステップ1101からの処理に戻って処理を続ける（ステップ1107）。

【0058】（4）ステップ1107での判定で、電圧モニタレジスタの値が電圧保存レジスタの値より一定以上小さかった場合、クロックテーブルの変更可能な各パラメータの値の2の商を求め、現在の値と置き換える。この結果、図4に示すクロックテーブルは、図12に示すような状態となる。この場合、システムのパラメータ欄406の値、及び、ディスプレイのクロック項目405の値は変更不可能な値であり、第1、第2のアプリケーションのパラメータ欄の値が1201、1202に示すように変更され、最大許容量のパラメータ欄1204の値が2の商に変更され、総和値の欄1203が再計算されている（ステップ1108）。

【0059】（5）この後、ステップ1108で再計算されたパラメータ値のうち、1つでも0があるか否かを判定し、ある場合、アプリケーションが必要としているMPUのクロック項目の値の和が0、または、メモリ制御ブロックのクロック項目404の値が0であった場合、そのアプリケーションが動作不可能となったと判定して、そのアプリケーションを終了してここでの処理を終了する（ステップ1109、1110）。

【0060】（6）ステップ1109の判定で値が0となったパラメータがなかった場合、PLL、分周器に対するコントロールレジスタの値を更新してセットし、現在の電圧モニタレジスタの値を電圧保存レジスタ1004に記憶し、その後、ステップ1101からの処理に戻って処理を続ける（ステップ1111）。

【0061】（7）ステップ1103での判定で、電圧モニタレジスタの値が電圧保存レジスタの値より一定以上大きかった場合、電圧モニタレジスタの値が最大値と

なっているか否かを判定する（ステップ1104）。

【0062】一般に、情報処理装置がバッテリ駆動されている場合、バッテリの電圧は、時間と共に低下していくので、前述までに説明したフローに従った処理が行われることになる。このため、ステップ1103での判定で、電圧モニタレジスタの値が電圧保存レジスタの値より一定以上大きかったという結果が得られる場合というのは、バッテリの交換が行われたとき、あるいは、バッテリが交流電源からフローティングされたようになったときである。

【0063】（8）ステップ1104の判定で、電圧モニタレジスタの値が最大値となっていた場合、低下していた電圧の回復が図られたことになるので、クロックテーブル内の変更可能なパラメータ値を2倍にして現在値に置き換え、ステップ1111からの処理を続ける。この場合、前述したステップ1108の処理でパラメータ値が少なくとも1/2とされているので、ここでこの処理で2倍としても問題を生じることはない（ステップ1106）。

20 【0064】（9）ステップ1104の判定で、電圧モニタレジスタの値が最大値となっていた場合、クロックテーブルの各パラメータを初期値に戻し、ステップ1111からの処理を続ける（ステップ1105）。

【0065】前述した本発明の第3の実施形態によれば、第1の実施形態の効果に加え、さらに外部電源の電圧変化に対して動作周波数を変更することが可能となり、システム全体の消費電力を押さえ、動作時間の延長を図ることができる。

【0066】図13は本発明の第4の実施形態による情報処理装置のハードウェア構成を示すブロック図、図14は必要とされるクロック周波数により電源電圧を変更するプログラムの処理動作を説明するフローチャート、図15は図14の処理で使用するクロッカー電圧テーブルの構成を説明する図であり、以下、これらの図を参照して本発明の第4の実施形態について説明する。図13において、1301はDC-DCコンバータ、1302は電圧設定レジスタであり、他の符号は図1の場合と同一である。

【0067】図13に示す本発明の第4の実施形態は、前述までに説明したクロックテーブル内の総和値のパラメータ欄の値に基づいて処理装置を構成する各ブロックに供給する電圧を変更することにより、処理装置の消費電力の低減を図ることを可能にしたもので、図1により説明した本発明の第1の実施形態に、DC-DCコンバータ1301と、電圧設定レジスタ1302とを設けて構成されている。DC-DCコンバータ1301は、電圧設定レジスタ1302の設定値を使用して電源電圧を決定し、図13に示している全ブロックへこの電圧を持つ電源を供給している。電圧設定レジスタ1302は、システムバス105を介して第1、第2のMPU10

1、102からアクセスが可能とされ、これらのMPUから制御可能とされている。

【0068】次に、図1-4に示すフローを参照して、必要とされるクロック周波数により電源電圧を変更するプログラムの処理動作を説明する。この処理では、図1-5に示すクロッカー電圧テーブルを使用する。図1-5において、1501はクロックテーブル内の総和値のパラメータ欄内の最大値、1502は対応する電圧設定レジスタ値である。

【0069】(1) この処理プログラムは、システムの起動時に起動され、起動されると、まず、図4により説明したクロックテーブルが更新されたか否かを判定し、更新が行われるまで、クロックテーブルが更新されたか否かの判定を繰り返す(ステップ1401)。

【0070】(2) その後、クロックテーブル内の総和値のパラメータ欄409の中から最大値を選択する。このパラメータ欄409の中での最大値は16であり、次に、図1-5に示すクロッcker電圧テーブルから前記最大値に対する電圧値を求める。この場合、図1-5で最大値16を超える総和値最大値1501の最小値が20となっているので、これに対応する電圧設定レジスタ値1502から得られる電圧レジスタ設定値は19となる(ステップ1402、1403)。

【0071】(3) 次に、ステップ1403で得た値が現在の電圧レジスタ設定値と同一であるか否かを判定し、同一の場合、ステップ1401の処理に戻って前述した処理を繰り返し、また、異なる場合、電圧設定レジスタに新しい値をセットする(ステップ1405)。

【0072】前述した本発明の第4の実施形態によれば、第1の実施形態の効果に加え、さらに動作周波数にあわせて電源電圧を変更することが可能となり、更なるシステム全体の消費電力を押さえることができる。

【0073】前述までに本発明の第1～第4の実施形態について説明したが、本発明は、これらの第1～第4の実施形態の複数を組み合わせて構成することもできる。

【0074】

【発明の効果】以上説明したように本発明によれば、動作するアプリケーションに対して充分な処理能力を実現でき、最小の消費電力で動作できるように、MPU等の機能ブロックを適正な周波数、作動電圧によって動作させることができ、情報処理装置の低消費電力化を図ることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施形態による情報処理装置のハードウェア構成を示すブロック図である。

【図2】アプリケーションプログラムの追加して起動したときに使用するプログラムの処理動作を説明するフローチャートである。

【図3】アプリケーションプログラム終了時に使用する

プログラムの処理動作を説明するフローチャートである。

【図4】図2の処理で使用するクロックテーブルについて説明する図である。

【図5】図2の処理でアプリケーションが追加起動された後のクロックテーブルの状態を説明する図である。

【図6】図3処理でアプリケーションが終了させられた後のクロックテーブルの状態を説明する図である。

【図7】本発明の第2の実施形態による情報処理装置のハードウェア構成を示すブロック図である。

【図8】ストリームデータの処理を開始するときに使用するプログラムの処理動作を説明するフローチャートである。

【図9】図8の処理でストリームデータの処理が開始された後のクロックテーブルの状態を説明する図である。

【図10】本発明の第3の実施形態による情報処理装置のハードウェア構成を示すブロック図である。

【図11】電源電圧が変動したときに動作周波数を変更するために使用するプログラムの処理動作を説明するフローチャートである。

【図12】図11の処理で動作周波数が変更された後のクロックテーブルの状態を説明する図である。

【図13】本発明の第4の実施形態による情報処理装置のハードウェア構成を示すブロック図である。

【図14】必要とされるクロック周波数により電源電圧を変更するプログラムの処理動作を説明するフローチャートである。

【図15】図14の処理で使用するクロッcker電圧テーブルの構成を説明する図である。

30 【符号の説明】

101、102 第1、第2のマイクロプロセッサー(MPU)

103 外部メモリ制御ユニット

104 外部メモリ

105 システムバス

106 機能ブロック

107 デジタル-アナログコンバータ(DAC)

108、118 第1、第2のフェイズロックループ(PLL)

40 111、113、115、120 第1～第4の分周器

117、122 第1、第2のコントロールレジスタ

701 ストリームデータ入力ブロック

1001 バッテリ

1002 電圧計

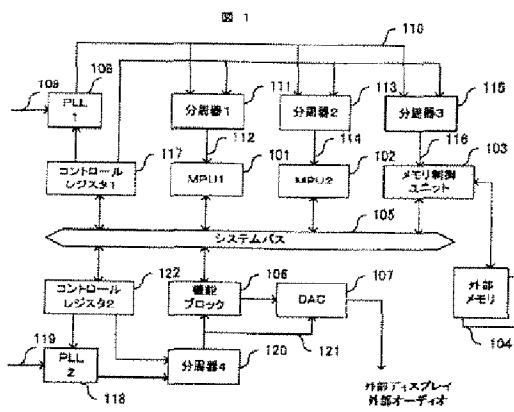
1003 電源モニタレジスタ

1004 電圧保存レジスタ

1301 D C-D Cコンバータ

1302 電圧設定レジスタ

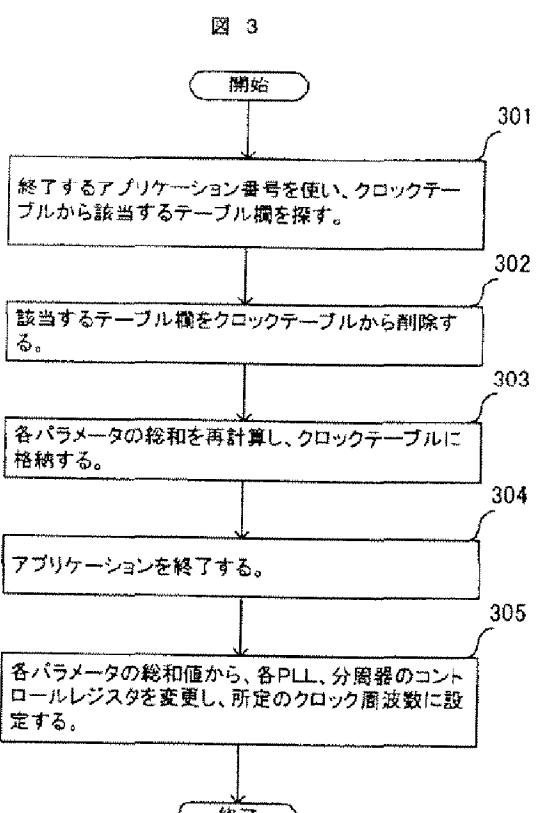
【図1】



【図4】

401	402	403	404	405
406 アプリケーション番号	MPU1	MPU2	メモリ	DAQ
407 0(システム)	1	1	1	-
408 1	5	15	5	15
409 2	7	0	2	-
410 録和値	13	15	8	15
411 最大許容量	20	20	10	15
412 判定	可能	可能	可能	可能

【図3】

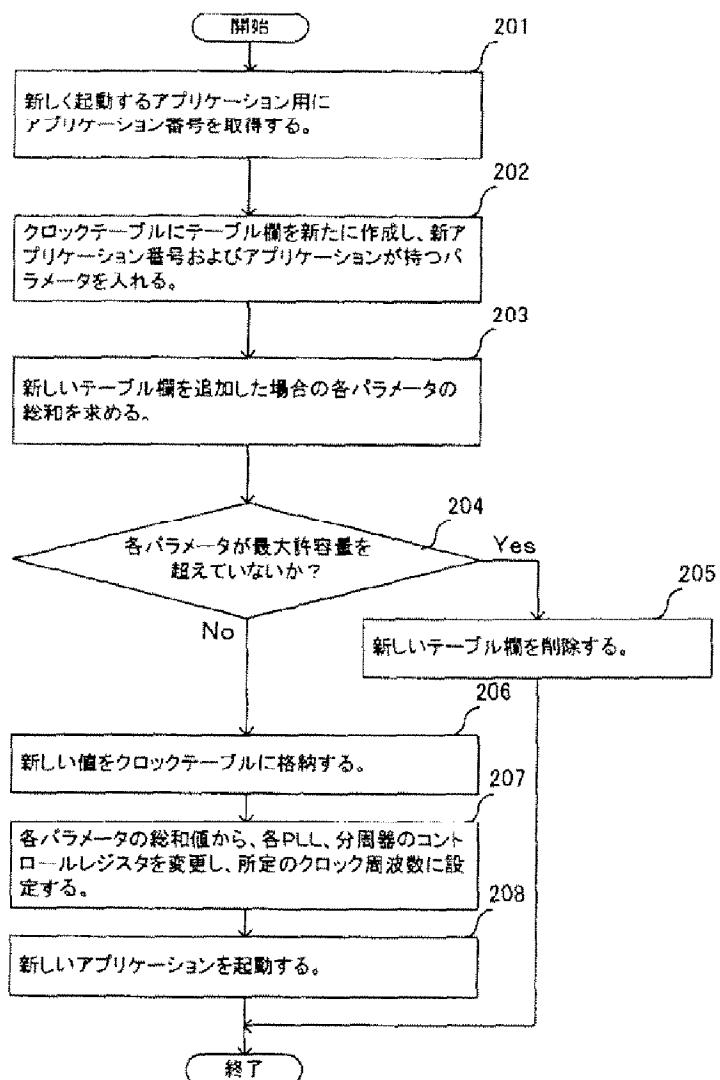


【図5】

401	402	403	404	405
406 アプリケーション番号	MPU1	MPU2	メモリ	ディスプレイ
407 0(システム)	1	1	1	-
408 1	5	15	5	15
501 2	7	0	2	-
502 3	3	1	1	-
410 録和値	16	17	9	15
411 最大許容量	20	20	10	15
412 判定	可能	可能	可能	可能

【図2】

図2



【図15】

図15

1501	総和許容量	10	15	20	15
1502	電圧指定レジスタ値	18	18	18	20

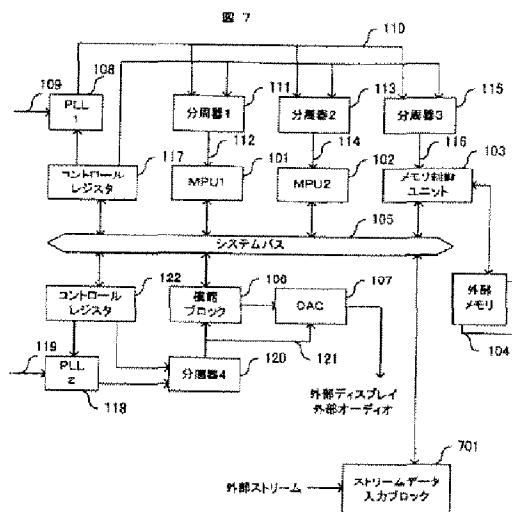
【図6】

401	402	403	404	405
アプリケーション番号	MPU 1	MPU 2	メモリ	ディスプレイ
0(システム)	1	1	1	—
1	5	15	5	15
3	3	1	1	—
飽和値	8	17	7	15
最大許容量	20	20	10	15
判定	可能	可能	可能	可能

【図9】

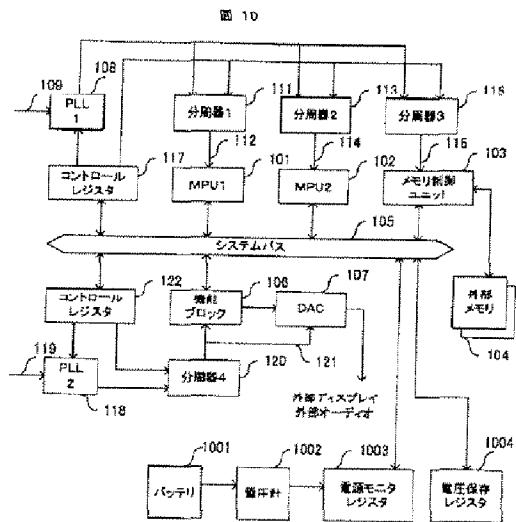
401	402	403	404	405
アプリケーション番号	MPU 1	MPU 2	メモリ	ディスプレイ
0(システム)	1	1	1	—
1#	5	10	2	8
2	7	0	2	—
トータル	13	11	5	8
最大許容量	20	20	10	15
判定	可能	可能	可能	可能

【図7】



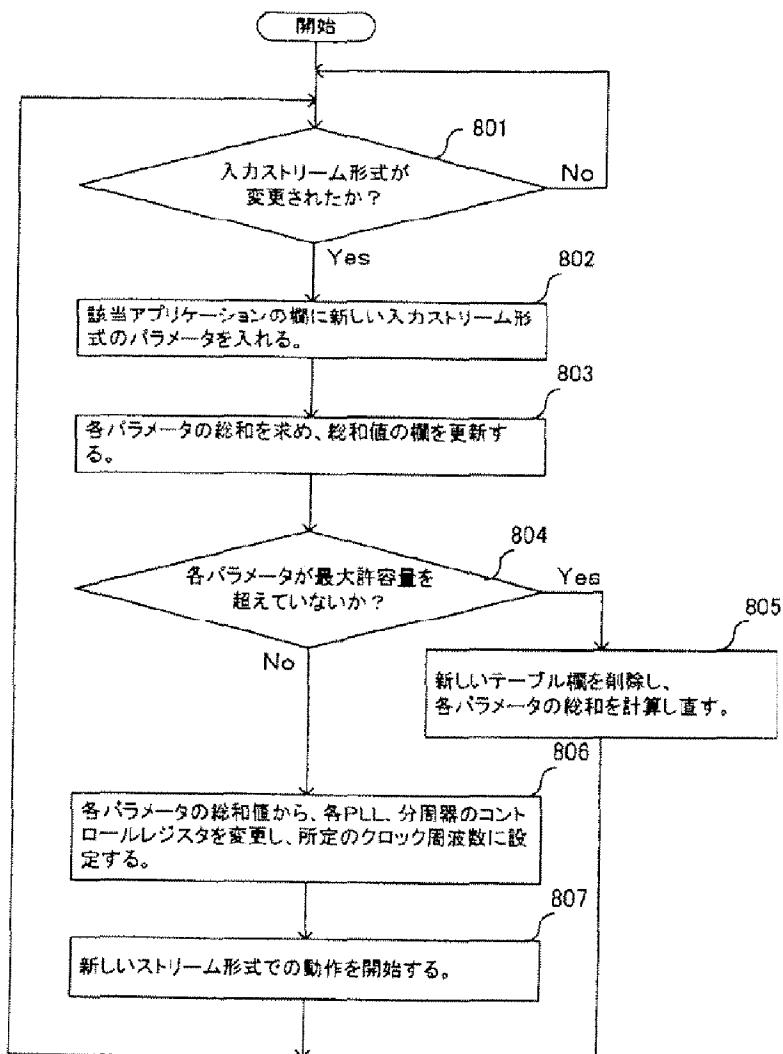
【図9】

【図10】



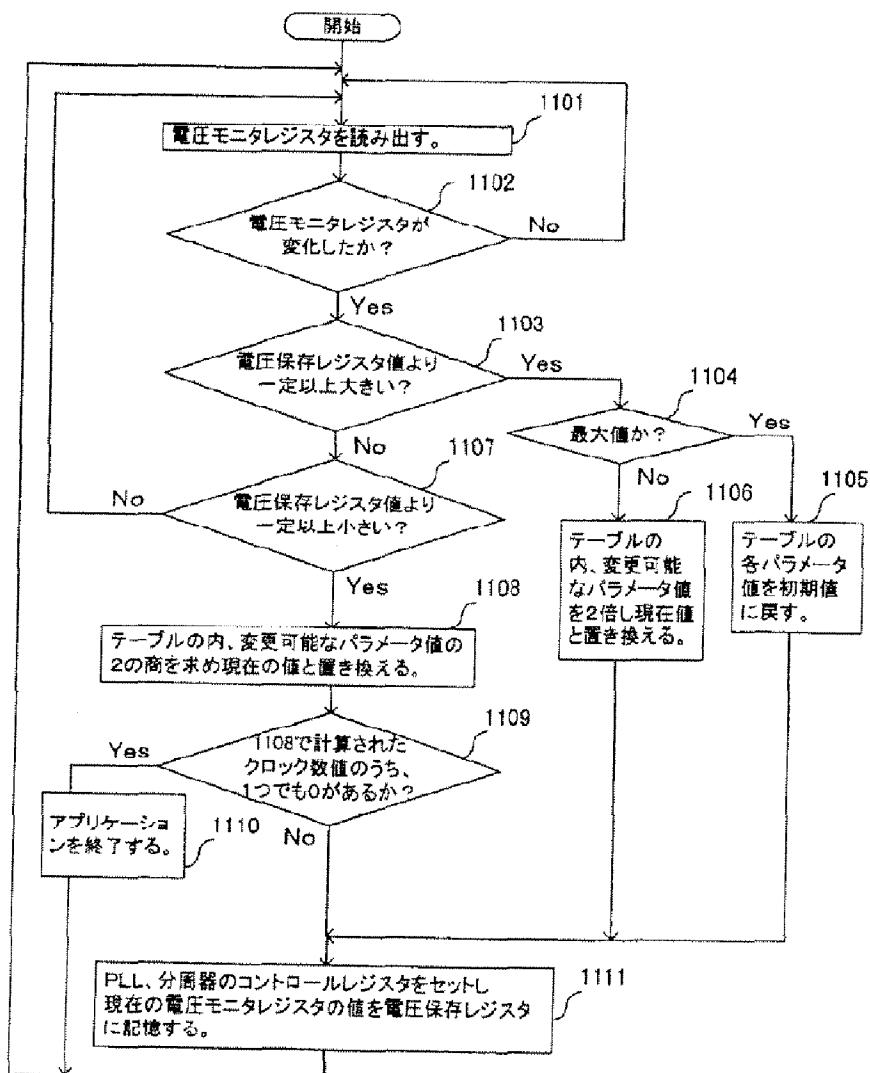
【図8】

図8



【図11】

図 11

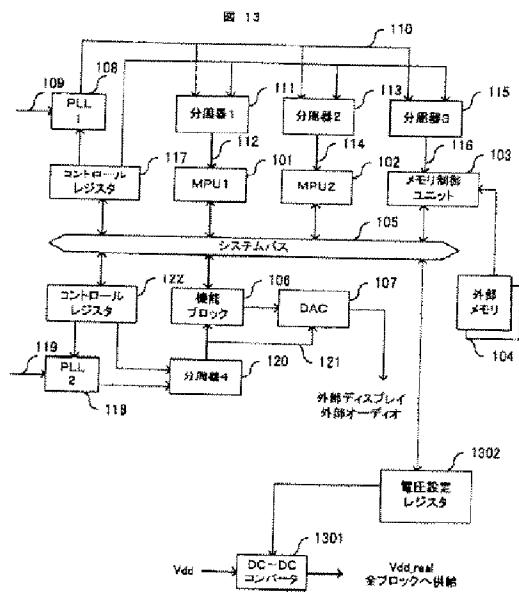


【図12】

図12

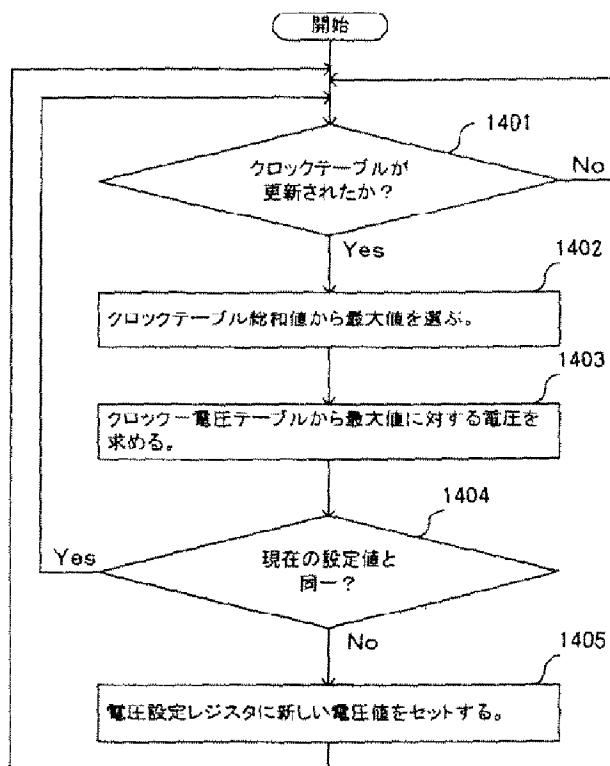
406	401 アプリケーション番号	402 MPU ₁	403 MPU ₂	404 メモリ	405 ディスプレイ
1201	0(システム)	1	1	1	—
1202	1	2	7	2	15
1203	2	3	0	1	—
1204	トータル	6	8	4	15
411	最大許容量	10	10	5	15
	判定	可能	可能	可能	可能

【図13】



【図14】

図 14



フロントページの続き

(72)発明者 中田 啓明
神奈川県川崎市麻生区玉禅寺1099番地 株
式会社日立製作所システム開発研究所内

(72)発明者 細木 浩二
神奈川県川崎市麻生区玉禅寺1099番地 株
式会社日立製作所システム開発研究所内
Pターム(参考) SB011 DA06 FF02 GG02 LL02 LL13
SB079 BA01 BC01